1986

The development of a knowledge-based system for information systems project development consulting /

Francis Matthew Lesusky
Lehigh University

THE DEVELOPMENT OF A KNOWLEDGE-BASED SYSTEM FOR INFORMATION SYSTEMS PROJECT DEVELOPMENT CONSULTING

by

Francis Matthew Lesusky

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of a

Master of Science

in

Industrial Engineering

Lehigh University

1986

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 5, 1986
(date)

John L. Wagner
Professor in Charge

L. E. Kane
Chairman of Department

ACKNOWLEDGMENTS

The author would like to acknowledge the contribution of Mr. R.L. Rhudy, Manager of Project Planning and Control, Business Information Systems, Air Products & Chemicals, Inc., for his time, enthusiasm, and "expert" advice in helping to develop the PROCON knowledge system. The author would also like to thank Dr. E.J. Miller, Manager of Knowledge-Based Systems, Management Information Services, APCI, for providing the INSIGHT software used to create PROCON.

The framework for this thesis was developed by Dr. J.C. Wiginton, Professor of Industrial Engineering, Lehigh University. The author acknowledges a debt of gratitude to Prof. Wiginton for his guidance in compiling this document and his advice on researching the project.

Additional thanks to the following members of Management Information Services, APCI, for participating in the demonstrations of the PROCON package: R.A. Branch, S.L. Haymon, R.H. Hoving, R. Kroll, C.R. Lewis, and W.A. Seibel.

Table of Contents

ABSTRACT

The foundation for a successful software development project is a well-conceived project plan which establishes the overall framework for effective project management. All aspects of the system development process must be addressed in a top-down manner to ensure that a global view of the total effort is taken at the outset of the project. Two major functions of planning an information systems development project are: 1) the selection of an appropriate development strategy, and 2) the assessment of risk associated with the development of the system.

This thesis deals with the development of a "knowledge-based system" to assist in choosing the most appropriate development approach to use when planning an information systems development effort, and the assessment of risk associated with that effort. The primary benefit of such a system is to have a wide range of knowledge and experience readily available for use by all project managers. The knowledge base that has been developed consists of heuristics regarding application development strategies and project risk along with rules from documented procedures for both subject areas.

The thesis also documents the procedure followed for building and refining the knowledge base and provides examples of its use in real-world situations.

I. Problem Statement

In managing a project to develop a business information system, several questions arise as to what type of approach is best to plan and control the project. For example, in the current Business Information Systems (BIS) environment at Air Products, there exists a choice of approaches that a project manager may select from:

1. Evolutionary development
2. Prototyping
3. Experiment (pilot)
4. Purchased software
5. End-user development
6. System Development Methodology - SDM/70

In order to help the BIS project manager decide which approach best suits his/her* application, he/she* consults the person who is most knowledgeable with the methodologies of the above mentioned development approaches. This 'expert' is the BIS Manager of Project Planning and Control (PPC). Through a series of question-and-answer consultations with the BIS project manager, the BIS PPC manager recommends which approach is the best fit for developing the application.

---

* "his/her" will be referred to hereafter as "his"
  "he/she will be referred to hereafter as "he"

This thesis deals with the automation of this consultation and decision-making process using a personal computer and currently available software (INSIGHT, developed by a Level Five Research, Melbourne Beach, FL). A knowledge-based system called the PROject CONsultant (PROCON) was built. It intended to be used by project managers to determine the type of development approach to use, and assess the degree of risk associated with the project.

Incorporated into the completed system are the rules which the BIS PPC manager uses to make his conclusions, drawn from:

1. Decision framework for selecting an appropriate development approach
2. Formal risk analysis questions and general rules-of-thumb
3. Prior experiences of project managers

The system leads the user (the BIS project manager) through a line of questioning such as:

"Is purchased software currently available?"

"What is the size of the project in resource hours?"

"What is the duration of the project in months?"

and, if appropriate, asks what his degree of confidence is that his response is correct.

Additionally, the system allows the user to choose a development approach (if he has already selected one), and pursue the line of questioning pertaining to that approach in order to support his selection. A similar process is used for performing a project risk assessment.

If the user needs more information about the question being asked the system provides a 'HELP' mode with more detailed information about the question.

When a conclusion has been reached at the end of the questioning process the user may request a hard-copy or on-screen report of all the questions and responses used to arrive at the conclusion.

II. Selection of Topic

The role of the PPC BIS Manager at Air Products is to provide consultation when preparing a strategy for development of business information systems, and also help to assess the risk associated with the project development effort. In addition to these functions, the BIS PPC Manager develops and maintains methods and procedures for BIS system development and support, administers the standard BIS development methodology (SDM) and develops or acquires various project control aids (such as the Project Manager Workbench).

A need has been identified to evaluate consistently the type of development approach to use in building information systems. Part of the PPC BIS manager responsibility is to consult with BIS project managers in the early stages of project planning to determine which approach should be used and also to perform a risk analysis for the project.

To help with this evaluation, the PPC BIS manager references a variety of sources in assessing factors common to most application development projects such as the type of technology to use, the type of skills needed, user attitude, time duration, business environment, and many others. Each new project is to be evaluated using the same criteria, with the PPC manager querying the project manager and finally arriving at a conclusion for development approach and risk analysis. A typical set of conclusions could be as follows:

I. Development approach: Purchased Software

Reasons: 1. Purchased software is available

2. Business environment where the software will be used is clearly defined

3. The user's knowledge of the application area is good

4. The scope/objective of the new the new system is clearly defined

II. Risk analysis:

1. Size risk -- low
2. Complexity risk - low
3. Technology risk - medium
4. Structure risk - medium

Since this approach to concluding a development/risk strategy is in the "diagnosis/prescription" format used in developing existing knowledge-based systems, it appeared to be a prime candidate for a similar approach. Two of the three main components were available:

1. An expert in the field - The BIS PPC manager and his knowledge of the subject matter along with his own rules-of-thumb for determinig development approach and project risk.

2. A knowledge base - The set of methods and procedures for project planning referenced when prescribing a development approach and assessing project risk.

The third component, an inference engine (software to encode the expert's rules), had not been determined at the time of the choice of the research subject area.

III. Background on Knowledge-Based Systems

Knowledge engineering is a branch of the computer science discipline known as artificial intelligence (AI). The other main branches of AI are (2) Natural Language Processing and (3) Robotics. Knowledge-based systems, (sometimes referred to as 'expert systems') began to be developed in the early 1970's with a business purpose in mind. That purpose was to capture as much available knowledge about a subject from an acknowledged expert, and to build a set of computer-readable rules that would simulate the decision-making strategies of the human expert in a consultation-type environment. That is, an interaction which proceeds by asking questions or providing information in a step-by-step process until a conclusion is reached.

An 'expert system' has been defined by Prof. Edward Feigenbaum of Stanford University:

> " ... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution ..." (6).

Most knowledge-based, expert systems use either the LISP (LISt Processing) or PROLOG (PROgramming in LOGic) computer programming

languages to write the logic rules of the systems. LISP (6), developed in the 1960's, combines list processing, recursive programming and extensibility (the ability to extend the language by defining new functions). PROLOG (6), developed in 1972, is a relational language that allows the knowledge engineer to describe details of the problem in a high-level (English-like) language based on first-order predicate logic.

The rules for PROCON are English-like expressions, as required by the "shell" program INSIGHT (developed using yet another programming language - PASCAL (11)). The English-like source code rules are translated by INSIGHT into PASCAL commands when the source code is "compiled" using INSIGHT's "preprocessor". However, the user does not need any knowledge of PASCAL to write rules for INSIGHT knowledge systems. That is, the implementation method is transparent and thus irrelevant to the user.

Knowledge-based systems were built originally to run on large mainframe computers. Eventually, highly specialized LISP-based machines assumed the task which formerly required to the mainframe. These machines, such as those offered by Symbolics Inc., were built specifically to run AI programs efficiently. With the increasing numbers and capabilities of personal computers, knowledge system vendors have built packages to run on such PC's. INSIGHT is one of these packages. Some others include:

ESP/ADVISOR, ExpertEase, and M.1.. The prices for PC-based knowledge system packages range from $100 to $5,000. By contrast, the prices for mainframe packages range from $23,000 to $80,000 (6).

The first knowledge-based system, MYCIN, was built at Stanford University in the mid 1970's. It is a diagnostic-type system designed to aid physicians in the diagnosis and treatment of meningitis and bacterimia infection. MYCIN runs on the DEC-20 computer and is a model for developing "diagnosis/prescription" systems (6). (PROCON uses a question/answer format similar to MYCIN).

IV. Expert Systems - Current and Future

Feigenbaum has discussed the current applications of expert systems and his observations are paraphrased here (4).

The current mode of operation for knowledge systems is one in which the user communicates with the system by keyboarding responses to questions at a computer terminal or PC keyboard. In the future, as voice recognition develops, the users will be able to speak to the computer instead of using a terminal for communication. Farther into the future it is likely that pattern-recognizing vision systems will enable users to respond pictorially.

Types of expert systems currently in use include:

1. Diagnosis and correction - what's wrong and how do we fix it?
2. Medical diagnosis and therapy - (MYCIN, PUFF)
3. Equipment failure and repair - (diesel/electric locomotives - G.E.)
4. Military application - strategies for battle plans based on situation assessment

5. Industrial operations

   a. order entry/order checking/scheduling of customer orders

   b. configuration of computers (used by DEC to configure VAX computers)

   c. optimum floor layout of computers (HITACHI)

   d. physical movement of mainframe computers from one location to another (IBM)

6. Nuclear power industry

   a. plant diagnosis

   b. crisis management

   c. safety systems

7. Financial industry

   a. insurance pricing of underwriting risk

   b. banks - assessment of the quality of loan applications

   c. accounting firms - audit planning

   d. assessment of consumer credit, loan applications

   e. structure of investment portfolios (risk - return)

V. Selection of Software

The software used to develop the knowledge system for this thesis is INSIGHT, version 1.2, developed by Level Five Research, Inc., Melbourne Beach, Florida. The reasons why this package was selected are as follows:

1. Cost - $95 (a version with extended capabilities is also available for $495).
2. Flexibility - INSIGHT allows the user to either search for an answer or pursue a specific line of questioning, and rules are written in English-like statements.
3. Availability - The INSIGHT software had previously been purchased by the Knowledge Systems Group of MIS at Air Products.

For an in-depth discussion of other knowledge system shells that were under consideration for developing PROCON, the following publications can be reviewed:

1. PC Magazine, April 16, 1985 (2),(3),(8),(15),(18)
2. "Expert Systems - Artificial Intelligence in Business" (6)

As a result of referring to these publications, the author was able to establish a framework for the selection criteria of a software package. The selection criteria included the following:

a. IF-THEN rules

b. Confidence factors

c. Backward-chaining

d. Depth-first search (pursue a goal)

e. Trace capability (audit trail)

f. HELP mode (EXPAND)

g. Prompted-menu display

VI. Building the Experiment

Given the choice of INSIGHT as the expert system shell, the next step in the research involved examining how conclusions are arrived at in a consultation between the MIS project manager and the BIS PPC manager. In this discussion, the expert described his question-and-answer process to arrive at a choice for a system development approach and to complete a risk assessment. Reference manuals for standard procedures and system development methodology also served as data to be compared and combined with rules-of-thumb that the BIS PPC Manager applies during his consultations.

The approach taken to building the rule set (INSIGHT program) was to combine interview data and standard procedures into INSIGHT rules. The goals (final conclusions) were set up to determine whether the user will choose to perform a risk analysis or choose the development approach to use. Within each of these main goals, subgoals were defined that follow the approach that the PPC BIS manager would use when consulting with a project manager. Depending on the level of detailed required to come to a final conclusion, the subgoals were defined to an even finer level of detail. Following is the list of goals that were defined:

main goal A. Development Approach

subgoals
- 1. Purchased Software
- 2. Evolutionary Development
- 3. Prototyping
- 4. Experiment (pilot)
- 5. End-user Development
- 6. Traditional Development (SDM/70 methodology)

main goal B. Project Risk

subgoals
- 1. Size Risk
- 2. Complexity Risk
- 3. Structure Risk
- 4. Technology Risk
- 5. Organizational impact risk

Generic rules were written that could be applied to a line of questioning for any of the goals listed. However, depending on how the user responded to the question that the rule was asking, any number of paths through the knowledge base could be taken to arrive at a conclusion. In some cases, intermediate goals (goals not specified as such) were embedded within the code in order to force a specific line of questioning to be pursued.

This concept can be illustrated by the manner in which the line of questioning was built for the development approach "Purchased Software". The rule set is as follows:

```
(INTERMEDIATE CONCLUSION 1)
    RULE For purchased software feasibility
    IF   purchased software is commercially available
    THEN purchased software may be feasible

(INTERMEDIATE CONCLUSION 2)
    RULE For purchased software feasibility
    IF   purchased software may be feasible
    AND  scope/objectives definition is thorough
    AND  user knowledge in application area IS good
    AND  user procedural change is feasible
    THEN purchased software is feasible CONFIDENCE 90
```

(FINAL CONCLUSION FOR PURCHASED SOFTWARE)

RULE For selecting Purchased Software

IF purchased software is feasible

AND definition of business environment is good

THEN MIS coordination is complex CONFIDENCE 90

AND Purchased Software should be used CONFIDENCE 90

AND Non-traditional development approach should be used

AND Development approach can be determined

The rule for INTERMEDIATE CONCLUSION 1 asks if purchased software is commercially available for the proposed project. If the user answers "Confident that it is true" to this question, he is further asked the question relating to feasibility of purchased software - INTERMEDIATE CONCLUSION 2. The questions asked for this rule set relate to project scope, user knowledge, and user procedural change. If the user answers "Confident that it is true" to all those questions, then it can be concluded (although not specified as a goal) that the Purchased Software alternative is, at this point, a possible solution. In order for Purchased Software to be concluded with 90 percent confidence, further questions are asked in the FINAL CONCLUSION that relate to business environment and MIS coordination. If the user responds "Confident that it is true" to these questions, then it can be concluded with 90 percent confidence that Purchased Software should be used as a development approach for this project.

A. Building the code -

The rules were written in a "building block" manner, that is, rules that would be "fired" first. These rules asked questions relating to project size, project duration, number of system interfaces, and so forth. The conclusions (the "THEN" part of these rules) became the "IF" part of more complex lines of questioning, building a more complex set of rules until all the list of final conclusions (GOALS) was exhausted. This line of rule development provided the BIS PPC manager with a logically organized structure of questioning that he was subsequently able to apply during his consultations while PROCON was still in a test mode.

For example, a typical set of conclusions that needed to be reached in order to advise a project manager to use Evolutionary Development would be:

> "In order to conclude that Evolutionary Development should be used for this project, the following questions relating to this goal must be asked:

What is the project size in manhours?

What is the project durations in calendar months?

What is the degree of integration complexity

(system interfaces)?

What degree of coordination of effort is needed

among various user groups?"

Since the lowest level rules were already written on a generic level for project size and project duration, it was necessary only to write rules specific to Evolutionary Development for integration complexity and coordination of effort with users.

Another responsibility of the BIS PPC Manager is to help with estimating the scope of the project. Currently there are manuals which the PPC Manager refers to when undertaking this task. In turn, when a project's scope is determined, it is also necessary to assess the risk associated with the project based on that scope. In PROCON a numeric value is input when information is requested for project size in terms of manhours, project duration in terms of months, number of system interfaces, number of users affected by the proposed system. Depending upon what numeric value is given, PROCON assigns an "impact" value to it, either "high", "medium", or "low".

In addition to the numeric values associated with impact factors, there are also other types of conclusions that are assigned impact

factors when assessing risk, such as: project management impact, geographic coordination impact, environment impact, project definition impact. These impact factors are also assigned values of "high", "medium", or "low". The impact factors have a direct correlation to numeric "weight-factors" in the estimating procedures included in the BIS PPC manuals and were put into PROCON in an effort to more closely integrate this package with the previously-existing manual estimating procedures.

B. Background on CONFIDENCE and THRESHOLD levels -

The coding process continued in the above manner until all goal rule-sets were written. The next task was to assign confidence factors for each conclusion. Confidence factors, as the term implies, refers to the degree of confidence that a given conclusion is true based on the answer given to the questions leading to that conclusion. The confidence option may be "turned on" by explicitly using the INSIGHT command CONFIDENCE ON, otherwise the default is CONFIDENCE OFF, in which case the conclusion is either 100% true or 0% true (i.e. false).

When confidence is turned ON, the user is asked to select a degree of confidence that each answer is true, based on the following criteria: (11)

Confident it is true	= 100%
Possibly true	= 75%
Not sure	= 50%
Possibly false	= 25%
Confident it is false	= 0%

Coupled with the confidence factor is a THRESHOLD statement which is used to evaluate each conclusion as to whether it is at the lowest rule level or the highest goal-rule-level. If the confidence of a

conclusion drops below the THRESHOLD level set by the knowledge engineer, the current line of questioning for a goal is abandoned and the next goal is evaluated.

Assignment of confidence factors is done by the BIS PPC manager based primarily on how comfortable he felt with the conclusion. This is a subjective approach for assigning confidence, but it is based upon the BIS PPC manager's knowledge of the subject matter and prior experience as a project manager.

The confidence factors of INSIGHT relate closely to the real-life situation that the BIS PPC manager faces when consulting with a project manager. In order to make the confidence factors as meaningful as possible, it was necessary to make several test (calibrating) runs through PROCON until it was agreed that the confidence factors truly reflected a real-life consultation.

C. EXPAND - PROCON'S Help Facility -

Since the PROCON package will eventually be made available to project managers to be used without the assistance of the BIS PPC manager, an essential part of the system is the HELP facility (the EXPAND statement in INSIGHT terminology). This feature allows the knowledge engineer to provide as detailed an explanation as necessary for the user to understand what is being asked for in a question. For example, a question relating to the number of interfaces that the proposed new system will have could be unclear to a first-time project manager. Using the EXPAND feature, PROCON would handle a query for more information in this way:

IN ORDER TO FIND OUT

the degree of integration complexity

WE FIRST NEED TO KNOW

the number of system interfaces

PROCON is asking for a positive numeric value to be entered, but if the user needs more information about integration complexity or system

interfaces, he can use the EXPAND option to receive a more detailed explanation. Specifically:

MORE INFORMATION ABOUT .....

number of system interfaces

Integration complexity refers to the number of already existing or new systems that will interface with the system being evaluated. The ratings are as follows:

complex = more than 1 system interface

average = 1 system interface

simple = 0 system interfaces

After reading the explanation, the user continues the line of questioning (presses the ENTER key) and will again be prompted for an answer to the integration complexity question.

As expertise was gained with building the knowledge system, the number of EXPANDs increased. Allowing BIS project managers to test PROCON as it was being built helped to identify and more clearly define the terminology that required more of an explanation in the form of EXPAND statements.

VII. Running the Experiment

To arrive at a deliverable finished product, extensive testing was done by both the "expert" and the knowledge engineer. The purpose of this rigorous testing was to ensure that the PROCON package truly mirrored the results of a real-world consultation session between the PPC Manager and the project manager.

Once the initial computer code was written using the INSIGHT software, several types of diagnostic tests were run against PROCON. The criteria for these tests was as follows:

1. The system should be able to provide for as many conclusions as possible
2. Ensure that the hierarchy of "rule-firing" was followed, that is:

   Rules were coded in a building-block manner;

   questions for primary conclusions were asked first,

   followed by intermediate conclusions, and ending with

   final conclusions (which ended the consultation)
3. Provide for as many "generic' rules as possible, where the same rule could be fired regardless of whether the consultation was for development approach or risk assessment.

4. Ensure accuracy of conclusions by analyzing results against the "real-world" environment, adjusting confidence levels where necessary.

The above procedure was followed for each final conclusion (goal), and allowed the knowledge engineer to "tune" the system so that what was arrived at for a conclusion truly reflected the real world.

A THRESHOLD of 60 was originally set for testing the CONFIDENCE levels of the conclusions. After a few adjustments of the THRESHOLD parameter to test which line of questioning PROCON should take, this level was felt to be the most accurate. Other THRESHOLD levels tested were 80, which led to not enough final conclusions being reached. Even though being 80 percent confident about a conclusion in the real world is a desirable level, it did not give the user the opportunity to reach a conclusion at a lower but still acceptable level of confidence. Also tested was THRESHOLD 50, which was felt to be too low since it could be inferred from a CONFIDENCE of 50 that the conclusion had just as much chance of being wrong as it did of being right.

VIII. Narratives of Real-Life Testing

A. Case 1:

Previous testing had been done by drawing upon the knowledge engineer's own experience in building information systems as well as that of the PPC BIS Manager and some previous consultations which he had done. The true test was to see how well PROCON performed in a real situation. In this test, the BIS PPC Manager had just consulted with a project manager on what type of development approach to use. From that consultation, he was able to draw a conclusion and advise the project manager that the most appropriate type of approach would be that of "building a Prototype" . His reasons for selecting this approach derived mostly from the methodology manuals used in consulting, but also from some heuristics (which were built into PROCON).

The requirements of the proposed system were that it would be a project of medium size, using both on-line and batch processing, with the users being willing to accept a prototyping approach, realizing that the prototype could turn into a full-scale system. The experience level on the MIS team that was assembled to work on the project was also conducive to using this type of approach.

Following his session with the project manager for the above system, the BIS PPC manager proceeded to use PROCON to see if it would draw the same conclusion that he drew for recommending a development approach. The first pass through PROCON led to "insufficient information to draw a conclusion". Investigating the reasons for not reaching a conclusion the first time, it was found that more rules were needed in PROCON to arrive at the same conclusion. It was, however, interesting to note that although a final conclusion was not reached, some intermediate conclusions were reached. Following are the intermediate conclusions:

The project size has been determined to be medium
The size impact relating to risk analysis is medium
The type of processing is online only
The user attitude toward MIS is good
The user organization is receptive to prototyping
User procedural change is expected to be minimal
User knowledge of the application area is good
The project appears to be suitable for prototyping

These intermediate conclusions were pointing PROCON in the direction of Prototyping. Specifically, the rule necessary for arriving at the Prototyping conclusion related to the application involving batch as well as online processing. Once that rule was in place, PROCON was able to conclude with 90 percent confidence that Prototyping was the approach to take.

B. Case 2:

The next case examined also dealt with determining a development approach. The project evaluated was considered to require an extensive labor effort by Air Products standards, with final implementation to take place over the course of a three-year period. The finished system would eventually have 5 "reporting" subsystems that would extract information from its database. In addition to the above, from the operational side of development, the users of the system were located at several sites throughout the U.S. This matter also had to be evaluated since a coordination of effort was required in order to ensure that all user requests were considered.

Given the parameters above, PROCON was able to conclude with 95 percent confidence that an Evolutionary Development approach should be taken to develop the system. The main components that pointed the diagnosis in the direction of Evolutionary Development were:

1. Project size was large
2. Project duration was long
3. Integration impact was high

C. Case 3:

The third real-life situation involved a risk analysis of the project development scenario described in Case 2.

Since the size of the project was large, PROCON inferred that the "size impact" was high. This was coupled with the response to the question concerning project duration which was determined to be long by APCI standards, therefore the combined size/duration impact on the project was inferred to be "low" (meaning that although the total manhours for the project is long, there should be sufficient time to complete the work). The project under evaluation will interface with two other systems, one which already in exists and another system which will be developed (the other system to be developed has been determined to be a project of medium risk). The MIS scope of the project has been thoroughly examined and the business environment that the new system will operate in has also been thoroughly investigated. The new system will use hardware and software technology currently available at Air Products. It should also be noted that user rapport with MIS has been good, and their expectations of what the capabilities of the new system will be have been clearly defined.

Given the above scenario, PROCON was able to conclude with 80 percent confidence that project risk can be determined. The evaluation summary is listed below:

1. Size risk is high, determined by
   a. project size large

2. Structure risk is low, determined by
   a. thorough scope definition
   b. thorough business environment definition
   c. well-defined user expectations
   d. impact on user environment is low
   e. novelty impact due to new technology is low

3. Technology risk is low, determined by
   a. hardware impact is low
   b. software impact is low

4. Organizational impact risk is low
   a. user attitude toward MIS is good
   b. user commitment is good
   c. user procedural change is feasible

IX. Development Considerations

The amount of time required to develop this knowledge system involves several areas that needed to be estimated. Following is a discussion of what areas were evaluated:

1. The expert's time to build PROCON into a working, but incomplete model. In this phase the goal structure was finalized and 95 percent of the rules were written.

2. The knowledge engineer's time to discuss the goals/rules with the expert and gain an understanding of the subject area.

3. The knowledge engineer's time to code and enter the rules, debug the system, add EXPAND code and fine-tune the system.

4. Software selection time used by the expert and knowledge engineer was devoted to learning about knowledge systems in general and reading trade publications which evaluated PC knowledge shells currently on the market. Also included in this time estimate was the amount of time spent evaluating the INSIGHT package to determine whether it would be the development tool for building PROCON.

5. There was no hardware selection to make since the knowledge engineer and expert had decided that the package chosen had to be able to run on the hardware that was currently available to them.

6. The learning curve to transfer the knowledge of maintaining PROCON, from the knowledge engineer to the expert, involved the following steps which the expert had to perform:

    a. Read the INSIGHT user's guide
    b. Understand and actually work with the syntax of the INSIGHT code
    c. Learn to use the PC text editor
    d. Understand and work with the structure of rules that were built into PROCON

7. Once the expert had performed Step 6, he spent additional time fine-tuning the rules that were in place for PROCON and adding new rules to the system.

X. MIS Management Response

After the PROCON knowledge system had been built, several demonstrations of it were presented to MIS management. The audience for the demonstrations included members of Business Information Systems (BIS) and the Knowledge Systems groups of MIS. In order to prepare two of the BIS groups for the demonstration, a "primer" of what questions might be asked by PROCON was given to one participant of each group. This participant actually performed a "live" consultation session with PROCON. Following is the "primer" questionnaire:

PROCON DEMO PRIMER

1. Size of project in MIS hours (numeric)
2. Duration of project in calendar months (numeric)
3. Number of subsystems generated by the application (numeric)
4. Project dependency
   a. not dependent on another project
   b. dependent on a high risk project
   c. dependent on a medium risk project
   d. dependent on a low risk project
5. Scope/objectives definition
6. User coordination, i.e. users are: (numeric)
   a. decentralized
   b. number of user orgs >= 3
   c. number of user orgs = 2
   d. number of user orgs = 1
7. Size of user organization (number of people) (numeric)
8. User knowledge of the application area
9. User understanding of information systems
10. User attitude toward MIS
11. User expectations
12. User management commitment
13. User procedural change that the new system will bring
14. Hardware impact
15. Software impact
16. MIS interdisciplinary considerations

a. development team is decentralized (non-TTown)
b. development team is BIS/DRM plus other MIS groups
c. development team is BIS/DRM only

17. Number of system interfaces (numeric)
18. Application logic complexity
19. Database status (if applicable)
    a. an existing database will be modified
    b. a new database will be developed
20. Type of processing involved
    a. Batch only
    b. Online
    c. Distributed
21. MIS project manager experience
22. MIS functional area experience
23. MIS technological experience

Following the demonstration, an evaluation sheet was given to each of the people attending the demonstration. The purpose of the evaluation was to:

1. Find out if this type of technology had a practical use in BIS at Air Products.
2. Since the participants were all project managers at one time, the evaluation of how PROCON worked was helpful in pointing out some of the gaps in logic that needed to be filled in before it was released as a usable product to current project managers.

Figure 1 shows the follow-up questionnaire that was used.

PROCON QUESTIONNAIRE

1. What is your prior exposure to knowledge systems?

2. Do you feel that the PROCON project was worth the time spent developing it?

3. Did PROCON perform better or worse than your expectations?

4. What capabilities of PROCON did you like most?

5. What capabilities (or lack of them) did you see as areas for improvement?

Figure 1

Following is a summary of the responses to the PROCON follow-up questionnaire:

Question 1:

In general, exposure to knowledge-based systems by the participants of the PROCON demonstrations was either non-existent or only in the form of reading magazine articles.

Question 2:

The general feeling was that the time spent to develop PROCON was worth the effort. The application can be used in real-life situations and BIS acquired a degree of expertise in developing knowledge-based systems. It should also be noted here that the original premise to build a knowledge-based system to determine development approach and assess risk is not a high priority in the current BIS environment.

Question 3:

Response to this question varied. Some of the replies indicated that PROCON performed better than originally anticipated, while others indicated that it was either equal to the level of expectation or that it fell short (in the area of the "help" mode) of expectations.

Question 4:

The capabilities of the system that people were most impressed with were those functions that were built into the INSIGHT software, namely:

Asking to respond to a question with a degree of confidence

Audit capability ("REPORT" at the end of a session)

Ability to process "unknown" responses

EXPAND mode

Pressing the tab key to highlight the desired response

Question 5:

Areas of improvement for PROCON were indicated by the following responses:

Ability to find out "why" a particular question is being asked

Needs more "HELP" screens

PROCON should let you know when it has abandoned a given line of questioning and rejected a particular development approach

The question that the user is supposed to answer should appear at the top of the screen and the area it relates to should appear below (INSIGHT displays it the other way around)

XI. Recommendations for a User's Guide

In order for PROCON to be considered a delivered products to APCI BIS, a user's guide must be written. Following is a list of topics that should be included in the user's guide:

1. Hardware/software configuration needed to run PROCON
2. Sign-on procedure
3. Description of goals
4. Brief description of how goals are evaluated
5. Description of THRESHOLD/CONFIDENCE
6. Description of PF keys
7. Sample run of a typical consultation session
8. Interpretation of results
   a. Conclusions drawn
   b. Audit trail
9. Copyright restrictions

XII. Expanding PROCON for the Future

The current version of PROCON (release 1.0) is currently a working system and is being used by the BIS PPC Manager as an "assistant" to his consultation sessions with project managers. The THRESHOLD of CONFIDENCE in this release has been set to 60, that is, a final conclusion will only be reached if PROCON has determined that the advice it gives is at least 60 percent certain of being true. While this is an acceptable level of confidence for consultations, it does not permit the BIS PPC Manager to examine all possible conclusions, regardless of the degree of confidence assigned to each.

As an expansion to the thesis work that allowed this system to be developed and become a functional part of his work environment, the BIS PPC Manager has identified some embellishments to release 1.0 of PROCON. Included in these embellishments are the capability of examining all possible conclusions for risk analysis. The current version of PROCON examines 5 categories for assessing risk on a project: size, complexity, structure, technology, and organizational impact. PROCON will allow a conclusion to be drawn if any three of these five categories can be successfully assessed - the assumption being that the project manager should know the degree of risk for most of the entire project. In the expanded version of PROCON (release 1.1), the BIS PPC Manager may also want to know if only two categories, or

even one, has been successfully assessed. He feels that he can then advise the project manager which areas need to be examined further before proceeding with another risk assessment.

It is envisioned that the approach taken for this enhancement to PROCON will be the development of additional "final" conclusions for risk analysis with all possible combinations of the five risk categories (currently it allows any combination of three categories). If only two categories of risk can be successfully assessed, then the CONFIDENCE factor will be 40. If only one risk category can be assessed, then the CONFIDENCE factor will be 20. In order for this analysis to take place at such a low degree of CONFIDENCE, the THRESHOLD must be set to a lower number than its current level of 60. The level, which has yet to be determined, would be no higher than 20. The structure for determining development approach is somewhat different than that for risk assessment, and theoretically would not be affected by resetting the THRESHOLD to a lower level.

Following the implementation of the upgraded risk analysis segment of PROCON, a major enhancement to the system will be the inclusion of an estimating module (hereafter referred to as the ESTImating CONsultant - ESTICON). This module will attempt to aid project managers in estimating the size and scope of a BIS development project. As with the development of PROCON, there are several methodology manuals

available in the BIS Project Planning and Control area for estimating projects. These manuals will form the basis, for rules which will go into ESTICON, however, a more diverse interviewing process will be conducted to capture the heuristic portion of estimating. BIS project managers will have the opportunity to contribute the benefit of their own estimating experiences as they relate to the standard BIS procedures for estimating. It is intended that this approach will result in a consultation package that will give project managers another automated tool for accurately estimating BIS development efforts.

XIII. Summary

Future knowledge-based systems will likely have a major impact in the area of training and in the daily performance of many job functions. Most procedures manuals currently used to document and communicate job performance information to employees can and will be replaced by small knowledge-based systems. The modularity of knowledge-based systems ensures that an "intelligent procedures manual" is readily available when needed and can be easily modified and updated as procedures change.

The development of PROCON demonstrates the business potential for knowledge systems with respect to centralizing, storing, and maintaining a knowledge-base representing an accumulation of project management knowledge and experience.

As the variables in project management strategies change, the rules contained within PROCON will also change. Entire segments relating to project estimating and productivity are planned as embellishments to the current PROCON system. As these enhancements are incorporated, the value of this product to the organization will increase and will promote its use, resulting in more robust project planning and estimating.

XIV. Location of PROCON Demonstration System

In compliance with thesis preparation guidelines, the following:

a. two PROCON system object code diskettes,
b. two PROCON source code diskettes,
c. PROCON program source listings,

are on file in the office of thesis advisor:

Professor John C. Wiginton
Department of Industrial Engineering
256a Packard Laboratory 19
Lehigh University
Bethlehem, Pa. 18015.

XV. References

(1) Chester, J. "Artificial Intelligence: Is MIS Ready for the Explosion", Infosystems, April 1985, pp. 74-78.

(2) Derfler, F. J. "An Affordable Adviser", PC Magazine, Vol 4 No 8, April 16, 1985, pp. 113-117.

(3) Derfler, F. J. "Expert-Ease Makes Its Own Rules", PC Magazine, Vol 4 No 8, April 16, 1985, pp. 119-124.

(4) Feigenbaum, E. "Texas Instruments Symposium on Artifical Intelligence", November 13, 1985.

(5) Fried, L. "Tools of the Manager's Trade: Expert Systems Enter the Corporate Domain", Management Technology, January 1985.

(6) Harmon, P. & King, D., Expert Systems - Artificial Intelligence in Business, John Wiley & Sons, 1985.

(7) Horwitt, E. "Exploring Expert Systems", Business Computer Systems, March 1985, pp. 48-57.

(8) Lemley, B. "Artificial Expertise: Intelligent Software For Problem Solving", PC Magazine, Vol 4 No 8, April 16, 1985, pp. 108-112.

(9) Level Five Research, INSIGHT User's Guide, 1985.

(10) Morley, R. & Taylor, W. Applications in Artificial Intelligence, Gould Electronics News Release, February 19, 1985.

(11) Morley, R. & Taylor, W. Applications in Artificial Intelligence, Gould Electronics News Release, March 4, 1985.

(12) Post, D.W. "Computers as Counselors", Business Computer Systems, January, 1985, pp. 58-60.

(13) Rauch-Hindin, W. "ARTIFICIAL INTELLIGENCE - A Solution Whose Time Has Come, Part 1", Systems & Software, December 1983, pp. 150-177.

(14) Rothfeder, J. "Is There Intelligent Life in the PC?", PC Magazine, Vol 5 No 1, January 14, 1986, pp. 139-148.

(15) Tello E. "The Languages of AI Research", PC Magazine, Vol 4 No 8, April 16, 1985, pp. 173-184.

(16) Ten Dyke, R. "Outlook on Artificial Intelligence", Journal of Information Systems Management (Fall 1984): 10-16.

(17) Webster, R. "A New Insight into EXPERT SYSTEMS", PC Magazine, Vol No 10, October 15, 1985, pp. 191-194.

(18) Webster, R. "Revealing Business Solutions", PC Magazine, Vol 4 No 8, April 16, 1985, pp. 125-130.

VITA

The author was born in Mahanoy City, Pennsylvania on June 19, 1953, the son of Leonard William and Sophie Sofranko Lesusky. Upon graduating from Mahanoy Area High School in June of 1971, he attended Bloomsburg State College (now Bloomsburg University), Bloomsburg, Pennsylvania. He graduated in May of 1975 with a B.S. in Mathematics. Upon graduation from Bloomsburg, he was employed at ComCom, Inc., Allentown, Pennsylvania where he was a commercial business systems programmer. Since October of 1979, he has been employed in the Management Information Services department of Air Products and Chemicals, Inc., Allentown, Pennsylvania, where he has developed and supported project management systems for corporate engineering as well as resource and maintenance management systems for corporate operations. He currently holds the title of systems analyst.